HYBRID OPTIMIZATION OF FEEDFORWARD NEURAL NETWORKS FOR HANDWRITTEN CHARACTER RECOGNITION

Wolfgang Utschick and Josef A. Nossek

Institute for Network Theory and Circuit Design
Technical University of Munich
Arcisstr. 21, 80333 Munich, Germany
Email: wout@nws.e-technik.tu-muenchen.de

ABSTRACT

An extension of a feedforward neural network is presented. Although utilizing linear threshold functions and a boolean function in the second layer, signal processing within the neural network is real. After mapping input vectors onto a discretization of the input space, real valued features of the internal representation of pattern are extracted. A vectorquantizer assigns a class hypothesis to a pattern based on its extracted features and adequate reference vectors of all classes in the decision space of the output layer. Training consists of a combination of combinatorial and convex optimization. This work has been applied to a standard optical character recognition task. Results and comparison to alternative approaches are presented.

1. INTRODUCTION

In [1] an extension of the Madaline Rule I algorithm of Widrow and Hoff [2] has been presented. The algorithm is related to a two-layer feedforward neural network consisting of adaptive neurons in the input layer and a boolean function (majority logic) in the second layer of the network. Because of the hard limiter activation function of the adaptive neurons and the binary properties of the boolean function there is no feasible gradient information and backpropagation like algorithms are not applicable. However in [1] has been shown that the principle of minimum weight disturbance applied to neural networks is an excellent alternative to error function approaches.

In this paper we present an extension of the proposed neural network. Although still utilizing linear threshold units and a boolean function, signal processing within the neural network is now real, because of the use of geometrical properties from internal representation of data. Moreover, embedding the binary decision space of the output layer into real space $\mathcal{R}^J$ makes implementing a vectorquantizer feasible. Figure 1 shows the architecture of the complete system. The

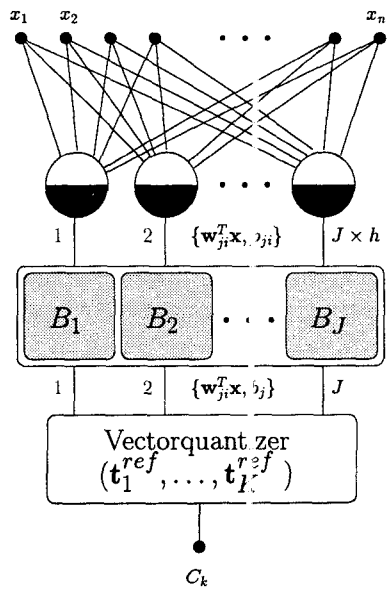

Figure 1: A feedforward neural network consisting of J parallel two-layer neural networks and a vectorquantizer. Each neural component shows binary properties, because of its boolean function $B_j(\bullet)$ in the second layer. A real valued output vector $\mathbf{o}_j(\bullet)$ of each input pattern is extracted. The vectorquantizer assigns a class C_k to the extracted feature vector of the second layer based on K reference vectors $\mathbf{t}_k^{ref}$.

complete training algorithm is given by a sequence of combined combinatorial and convex optimization problems. The objective of the training algorithm is a correct embedding of input pattern $\mathbf{x} \in \mathcal{R}^n$ according to their desired output target $\mathbf{t}_x^{ref}$ element of a sphere $\mathcal{S}^{J-1} \subset \mathcal{R}^J$. Embedding of pattern stands

0-8186-7919-0/97 $10.00

for an adjustable mapping of $\mathbf{x}$ into the domain of the boolean functions $b_j = B_j(\bullet)$, $b_j \in \{-1,+1\}$ and is composed, for each network j, by the subsequential mapping of $\mathbf{x} \mapsto l_{ji} = \mathbf{w}_{ji}^T\mathbf{x}$ into the space of local fields $\mathbf{l}_j$ of all $i = 1,\ldots,h$ hidden neurons of each input layer j and the hard limiter function $l_{ji} \mapsto p_{ji} = \mathrm{sgn}(l_{ji})$, $p_{ji} \in \{-1,+1\}$. The weight vector $\mathbf{w}_{ji}$ represents the weighted summation of all inputs of the neural network. For each subsystem $j = 1,\ldots,J$, elements of $\mathbf{p}_j \in \{-1,+1\}^h$ correspond to convex regions $\{\mathbf{x} \in \mathcal{R}^n \mid p_{ji} = \mathrm{sgn}(\mathbf{w}_{ji}^T\mathbf{x}) \ , \ i = 1,\ldots,h\}$ in the space of input pattern [2, 3, 4, 5, 6]. These regions are called cells z_j and are indicated by the decimal value of its binary equivalent, i.e. $z_j = \{p_{j1}p_{j2}\cdots p_{jh}\}_{\mathrm{dec}}$. The number of cells $|\mathcal{Z}_j| \leq 2^h$ in each neural component is finite. Therefore the mapping of the boolean functions may also be defined by means of a lookuptable. According to the internal representation of a pattern, i.e. euclidian distances to adjacent cells, a real output o_j of each component is extracted. The vectorquantizer's final decision is based on the minimal distance to a reference vector $\mathbf{t}_k^{ref}$ element of classes $C_1, C_2, \ldots, C_K$. In other words, the first layer of the neural network performs a mapping of the input pattern onto elements $z_j \in \mathcal{Z}_j$ of a discretization of the input pattern space. From individual internal representation of pattern, feature vectors are extracted and a vectorquantizer makes a decision. This interpretation makes the phrase embedding of pattern more clear. In the following, an outline of the training algorithm, i.e. the hybrid optimization, of the feedforward neural network is given. The system has been applied to optical character recognition tasks. Results and comparison to alternative classification systems are presented.

2. PROVIDING TARGETS FOR SUPERVISED TRAINING OF THE NEURAL NETWORK COMPONENTS

The proposed supervised training is sequential, i.e. the algorithm is iteratively applied to a very restricted number $r = 1, 2, 3, \ldots$ of pattern. This set of relevant samples is randomly drawn from the underlying training set and exclusively consists of misclassified pattern, i.e. pattern with incorrect output vector $\mathbf{o}(\mathbf{x})$, according to the input of the vectorquantizer and its given set of current reference vectors $\mathbf{t}_k^{ref}$. The objective of providing targets for the supervised training of the neural components of the system is to find a target

$$\mathbf{t}^{prov} := \mathbf{o}(\mathbf{x}) + \Delta\mathbf{t} \tag{1}$$

subject to

$$||\mathbf{t}^{prov} - \mathbf{t}_x^{ref}||_2^2 \quad < \quad ||\mathbf{t}^{prov} - \mathbf{t}_k^{ref}||_2^2,$$

where $\mathbf{t}_x^{ref} \neq \mathbf{t}_k^{ref}$ is the valid reference vector of $\mathbf{x}$. However, in respect to the principle of minimal weight disturbance in the next section, a convex optimization problem

$$\min_{\Delta\mathbf{t}} \frac{1}{2}\Delta\mathbf{t}^T\mathbf{C}\Delta\mathbf{t} \tag{2}$$

is given, subject to the tesselation of decision space by the set of reference vectors. The matrix $\mathbf{C}$ denotes a modified metric in target space, because the principle of minimal weight disturbance is related to weight space and there are different scales of $||\Delta\mathbf{w}_{ji}||_2$ according to $|\Delta t_j|$.

3. PRINCIPLE OF MINIMAL WEIGHT DISTURBANCE FOR EMBEDDING PATTERN

With the principle of minimal weight disturbance Widrow and Hoff left some flexibilty of possible realizations. In [1] a minimal weight change $\Delta\mathbf{w}_{ji}$ of the neurons of the input layer has been proposed for beeing a criterion of the embedding of multiple input pattern. For the set $\mathbf{x}_\mu$, $\mu = 1,\ldots,r$ of input pattern, each training step of the hidden neurons of the first layer according to provided targets is given as the combinatorial optimization problem

$$\{z_{1j}^*,\ldots,z_{rj}^*\} = \arg \min_{\substack{z_{\mu j} \\ \mu=1,\ldots,r}} \sum_{i=1}^{h} ||\Delta\mathbf{w}_{ji}^*||_2^2 \tag{3}$$

subject to $o_j(\mathbf{x}_\mu) = t_{\mu j}^{prov}$, i.e.

$$\begin{aligned} B_j(z_{\mu j}) &= \mathrm{sign}(t_{\mu j}^{prov}) \\ d(\mathbf{x}_\mu, \hat{z}_{\mu j}) &= |t_{\mu j}^{prov}| \end{aligned}$$

and its implicit convex optimization[1] for mutual robust embedding of all vectors $\mathbf{x}_\mu$, $r = 1,\ldots,r$,

$$\Delta\mathbf{w}_{ji}^* = \arg \min_{\Delta\mathbf{w}_{ji}} ||\Delta\mathbf{w}_{ji}||_2^2 \tag{4}$$

subject to

$$\begin{aligned} p_{ji} \cdot \Delta\mathbf{w}_{ji}^T \frac{\mathbf{x}_\mu}{||\mathbf{x}_\mu||_2} &\geq (|t_{\mu j}^{prov}| - p_{ji} \cdot \mathbf{w}_{ji}^T \frac{\mathbf{x}_\mu}{||\mathbf{x}_\mu||_2}) \\ \mu &= 1,\ldots,r, \end{aligned}$$

with $p_{ji} = \{z_{\mu j}^*\}_{\mathrm{bin},i}$ of (3) for each j. The constraints are derived by the claim for maximal robustness $|t_{\mu j}^{prov}|$ of embedded pattern, see also [7, 8]. The boolean function and the distance measure to adjacent cells provide the real valued output of each neural network by

[1] only for case of $\mathbf{t}^{prov}$ and $\mathbf{t}_x^{ref}$ beeing elements of equal hyperrectangles of $\mathcal{R}^J$.

$o_j(\mathbf{x}_\mu) = B_j(z^*_{\mu j}) \cdot d(\mathbf{x}_\mu, \hat{z}^*_{\mu j})$. Whereas the distance measure $d(\bullet)$ is given by the minimal distance of an input vector $\mathbf{x} \in z$ in space of local fields to a cell $\hat{z}$ with $B(\hat{z}) \neq B(z)$.

For reducing the computational complexity, the original combinatorial search for internal representation $z^*_{\mu j}$ of multiple pattern has been replaced by a suboptimal choice of cells $z^*_{\mu j}$ for each pattern. Therefore, the objective function of (3) is rewritten to $z^*_{\mu j} = \arg\min_{z_\mu} ||\Delta \mathbf{l}_j||_2^2$ subject to $p_{ji}(l_{\mu i} + \Delta l_{\mu i}) \geq ||\mathbf{x}_\mu||_2 \cdot |t^{prov}_{\mu j}|$ for $B_j(z_\mu) = \text{sign}(t^{prov}_{\mu j})$. Input $\mathbf{x}_\mu$ is strictly embedded into cell $z^*_{\mu j}$ with minimal distance $d(\mathbf{l}_{\mu j}, z^*_{\mu j})$ to the still misclassified vector $\mathbf{l}_{\mu j} = \mathbf{w}^T_{ji}\mathbf{x}_\mu$ in the space of local fields, i.e.

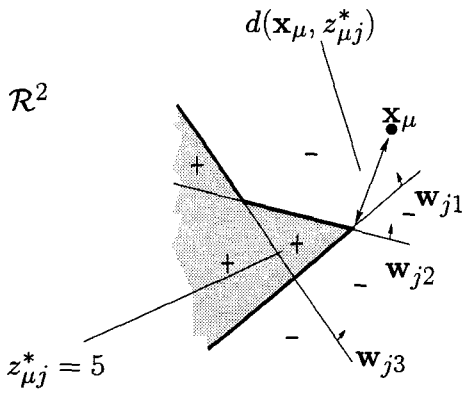

Figure 2: Discretization of the input space $\mathcal{R}^2$ with $h = 3$ hidden neurons in the input layer. $z^*_{\mu j} = 5 = \{+1\ -1\ +1\}_{\text{dec}}$ represents the cell with minimal distance $d(\mathbf{x}_\mu, z^*_{\mu j})$ to the misclassified input pattern $\mathbf{x}_\mu$ with target $\text{sign}(t^{prov}_{\mu j}) = +1$. Note, that distances between input vectors and cells are measured in the space of local fields $l_{ji} = \mathbf{w}^T_{ji}\mathbf{x}_\mu$.

$$z^*_{\mu j} = \arg\min_{z_j} \inf_{\mathbf{l}_j} ||\mathbf{l}_{\mu j} - \mathbf{l}_j||_2^2 \qquad (5)$$

subject to

$$\begin{aligned} B_j(z_j) &= \text{sign}(t^{prov}_{\mu j}) \\ \{z_j\}_{\text{bin},i} \cdot l_i &> 0, \end{aligned}$$

where $\{z_j\}_{\text{bin}}$ displays the binary equivalent of possible cell indices and $\inf_{\mathbf{l}_j} ||\mathbf{l}_{\mu j} - \mathbf{l}_j||_2^2 = d(\mathbf{l}^{\mu j}, z_j)$ for "$\mathbf{l}_j \in z_j$", see also figure 2. Thereby the complexity is dramatically reduced from $o(2^{r \cdot h})$ to $o(r \cdot 2^h)$ in number of convex optimization problems (4), whereby the distances $d(\bullet)$ are computed in linear complexity $o(h)$ of standard vector operations. After calculation (5) of cells $\mathbf{z}^*_{\mu j}$ the robust embedding (4) of all input pattern $\mathbf{x}_\mu$ according to $\mathbf{z}^*_{\mu j}$ is outperformed.

In [8, 9] the relevance of the introduced convex optimization problem to Structural Risc Minimization has been shown.

The nature of the complete algorithm may be described as follows: for the set of input pattern $\mathbf{x}_1, \ldots, \mathbf{x}_r$ and all hidden neurons $i = 1, \ldots, h$ of each subsystem $j = 1, \ldots, J$ find an embedding $\mathbf{z}^*_{1j}, \ldots, z^*_{rj}$ with $B_j(z^*_{\mu j}) = \text{sign}(t^{prov}_{\mu j})$ and $d(\mathbf{x}, \hat{z}^*_{\mu j}) = |t^{prov}_{\mu j}|$ according to minimal common costs of $||\Delta \mathbf{w}_{ji}||_2$ of all adaptive neurons and the objective of robust embedding $p_{ji} \cdot \mathbf{w}^T_{ji} \frac{\mathbf{x}_\mu}{||\mathbf{x}_\mu||_2} \geq |t^{prov}_{\mu j}|$.

Due to the random search of misclassified pattern the training algorithm shows stochastical properties and a monitoring of the process is important. Therefore, training is split into S training epochs. In each training epoch $s = 0, \ldots, S$ a test of the state of the network is performed periodically, i.e after a sequence of iteration steps[2] the number $\epsilon^{(s)}$ of misclassified pattern within the training set or a cross-validation set is calculated. All network parameters of the training sequence are saved if $\epsilon^{(s)}$ has improved.

4. LEARNING VECTOR QUANTIZATION

After each training epoch s an adaptation of the reference vectors $\mathbf{t}^{ref,(s+1)}_k := \mathbf{t}^{ref,(s)}_k + \Delta \mathbf{t}^{ref}_k$ for the set of all classes C_k, $k = 1, \ldots K$, is performed. The system of reference vectors $\mathbf{t}^{ref,(0)}_k$ originally stems from a 1-of-10 winner-takes-all code or any other encoding of multiple classes in decision space [10]. Various criteria are possible for adaptation of the reference vectors. Whereas in [11] a Fisher criterion is used for an unsupervised type of learning vector quantization, this work prefers an error criterion ϵ, i.e. the total number of misclassified pattern within the training set. The optimization problem is given by

$$\{\Delta \mathbf{t}^{ref}_k\} = \arg \min_{\substack{\Delta \mathbf{t}_k \\ k=1,\ldots,K}} \epsilon(\mathbf{t}^{ref,(s)}_k + \Delta \mathbf{t}_k) \qquad (6)$$

subject to

$$\begin{aligned} \mathbf{t}^{ref,(s)}_k + \Delta \mathbf{t}_k &\in \mathcal{S}^{J-1} \\ k &= 1, \ldots, K, \end{aligned}$$

i.e. minimizing the total number of misclassified vectors within the set of training pattern by means of a final adaptation of the reference vectors $\mathbf{t}^{ref}_k$ on a sphere, after each training epoch. For finding local extrema a newton based approach is applied. Gradients of $\partial\epsilon/\partial \mathbf{t}_k$ are calculated numerically.

[2] each step consists of optimization problems (2) - (4).

5. OPTICAL CHARACTER RECOGNITION

This work has been applied to an optical character recognition task. Recognition of handwritten digits $0-9$. The results are related to mixtures of NIST-Training and NIST-Test databases called MNIST, see also [12]. After a generalized Hough-Transformation for feature extraction [13], the feedforward neural network was trained. For each class $0,\ldots,9$, i.e. $J = K$, networks consist of $h = 3$ hidden neurons, each neuron fully connected with 194 inputs of extracted features, i.e. networks of the first layer have $5850 = 10 \times 195$ free parameters (bias of each neuron included). The number of training and test pattern is equal to [12]. Figure 3 presents a comparison of different classifier methods, partly published in [12, 8]. For rejection of 3.6% input pattern the misclassification error of the presented classifier lies about 0.5%. Note, that the network only requires 3×10 dot-products in the hidden layer for classification of a single character, whereas **SVM** requires more than 1000×10 dot-products [8]. The higher computational costs of LeNet classifiers in comparison with fully connected nets has been reported in [12].

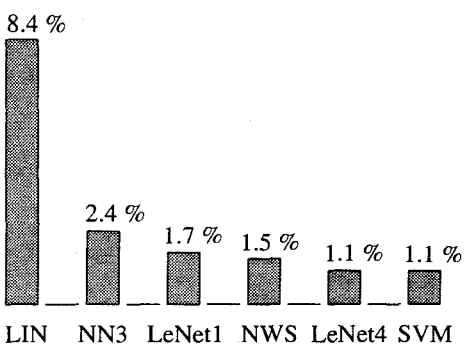

Figure 3: Comparision of the presented feedforward neural network (NWS) with a linear classifier (LIN), 3-nearest neighbor classifier (NN3), multilayer neural network (LeNet1,LeNet4) and a support vector machine (SVM).

6. CONCLUSION

In this paper a minimum weight disturbance principle for supervised training of feedforward neural networks has been presented. Additionally, the competitive relevance of feedforward neural networks based on hard limiter activation functions, combined with vectorquantization methods and without access to gradient based learning algorithms for the neural components has been demonstrated.

7. REFERENCES

[1] J.A. Nossek, P. Nachbar, and A.J. Schuler. Comparison of Algorithms for Feedforward Multilayer Neural Nets. In *International Conference on Circuits and Systems*, volume 3, pages 380–384. IEEE, 1996.

[2] B. Widrow and M.A. Lehr. 30 Years of Adaptive Neural Networks: Perceptron, Madaline, and Backpropagation. *IEEE Proc.*, 78(9):1415–1441, 1990.

[3] B. Widrow, R.G. Winter, and R.A. Baxter. Layered Neural Nets for Pattern Recognition. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 36(7):1109–1118, 1988.

[4] G.J. Gibson and C.F.N. Cowan. On the Decision Regions of Multilayer Perceptrons. *IEEE Proc.*, 78(10):1590–1594, 1990.

[5] W. Utschick and J.A. Nossek. Bayesian Adaptation of Hidden Layers in Boolean Feedforward Neural Networks. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 4, pages 229–233. IAPR, 1996.

[6] R. Eigenmann and J.A. Nossek. Constructive and Robust Combination of Perceptrons. In *Proceedings of 13th International Conference on Pattern Recognition*, volume 4, pages 195–199. IAPR, 1996.

[7] P. Nachbar, J. Strobl, and J.A. Nossek. The generalized adatron algorithm. In *International Symposium on Circuits and Systems*, volume 4, pages 2152–2156. IEEE, 1993.

[8] C. Cortes and V.N. Vapnik. Support-Vector Networks. *Machine Learning*, (20):273–297, 1995.

[9] V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1995.

[10] W. Utschick, H.-P. Veit, and J.A. Nossek. Encoding of Targets for Supervised Training of Neural Networks. to be submitted, 1997.

[11] R. Lengellé and T. Denœux. Training MLP's Layer by Layer Using an Ojective Function for Internal Representations. *Neural Networks*, 9(1):83–97, 1996.

[12] L. Bottou, C. Cortes, J.S. Denker, H. Drucker, I. Guyon, L.D. Jackel, Y. LeCun, E. Sackinger, P. Simard, V. Vapnik, and U.A. Miller. Comparison of classifier methods: A case study in handwritten digit recognition. In *Proceedings of 12th International Conference on Pattern Recognition and Neural Network*, 1994.

[13] W. Utschick, P. Nachbar, C. Knobloch, A. Schuler, and J.A. Nossek. The Evaluation of Feature Extraction Criteria Applied to Neural Network Classifiers. In *Proceedings of 3th International Conference on Document Analysis and Recognition*, pages 315–318. IEEE, 1995.